


ΕΠΛ222: Λειτουργικά Συστήματα
(μετάφραση στα ελληνικά των διαφανειών του βιβλίου Operating Systems: Internals and Design Principles, 9/E, William Stallings)

Ενότητα 1 (Κεφάλαιο 2) Εισαγωγικές Έννοιες

Οι διαφάνειες αυτές έχουν συμπληρωματικό και επεξηγηματικό χαρακτήρα και σε καμία περίπτωση δεν υποκαθιστούν το βιβλίο

Πάργος Α. Παπαδόπουλος
 Τμήμα Πληροφορικής
 Πανεπιστήμιο Κύπρου



1

Περιεχόμενα

- Στόχοι και λειτουργίες του Λ.Σ.
- Η εξέλιξη των Λ.Σ.
- Κύρια επιτεύγματα.
- Εξελίξεις που οδήγησαν στα μοντέρνα Λ.Σ.
- Microsoft Windows.
- Συστήματα UNIX.
- Linux.
- Android.

2

Λειτουργικό Σύστημα

- Ένα πρόγραμμα που ελέγχει την εκτέλεση των προγραμμάτων εφαρμογών.
- Μία διασύνδεση (interface) μεταξύ των εφαρμογών και της μηχανής.
- Βασικοί στόχοι ενός Λ.Σ.:
 - Παροχή δευκολύνσεων στη χρήση του Η/Υ.
 - Πιο αποδοτικά χρήση των διαθέσιμων πόρων (resources) του συστήματος.
 - Δυνατότητα για εύκολη επέκταση με επιπλέον λειτουργικότητες, μέσω της ανάπτυξης, ελέγχου και εισαγωγής τους χωρίς να επηρεάζεται η λειτουργία του συστήματος.
 - Δυνατότητα διόρθωσης υπαρχόντων λαθών.

3

Η θέση του Λ.Σ. στον Η/Υ

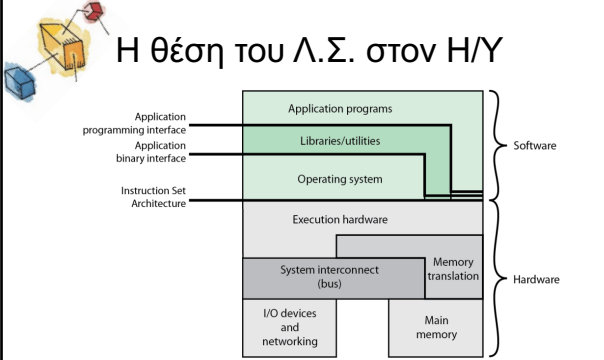


Figure 2.1 Computer Hardware and Software Infrastructure

4

Υπηρεσίες που παρέχονται από το Λ.Σ. — 1

- Ανάπτυξη προγραμμάτων.
 - Όπως κειμενογράφοι (editors) και απασφαλματωτές (debuggers) που αν και δεν είναι μέρος ενός Λ.Σ. συνήθως παρέχονται μαζί με αυτό ως χρήσιμα προγράμματα (utilities).
- Εκτέλεση προγραμμάτων.
 - Χρονοδρομολόγηση (scheduling) των ενεργειών που απαιτούνται για την εκτέλεση ενός προγράμματος (π.χ. φόρτωμα κώδικα και δεδομένων στην κύρια μνήμη, άνοιγμα αρχείων, κλπ.).
- Πρόσβαση σε συσκευές Εισόδου/Εξόδου.
 - Κάθε συσκευή έχει διαφορετική διασύνδεση με το σύστημα, εκτελώντας το δικό της σετ εντολών.
 - Το Λ.Σ. παρέχει μία τυποποιημένη διασύνδεση στους χρήστες, έτσι ώστε με απλές γενικευμένες εντολές να μπορούν αυτοί να ελέγχουν τις συσκευές.

5

Υπηρεσίες που παρέχονται από το Λ.Σ. — 2

- Ελεγχόμενη πρόσβαση σε αρχεία.
 - Πρόσβαση σε διαφορετικού είδους δεδομένα μέσω μίας κοινής διασύνδεσης με τους χρήστες.
 - Παροχή ασφάλειας σε συστήματα Η/Υ με πολλούς χρήστες με ελεγχόμενη πρόσβαση στις πληροφορίες των χρηστών.
- Πρόσβαση στον Η/Υ.
 - Ελεγχόμενη πρόσβαση στο σύστημα και τους διαθέσιμους πόρους.

6

Υπηρεσίες που παρέχονται από το Λ.Σ. — 3

- Ανίχνευση σφαλμάτων και αντιμετώπισή τους.
 - Εσωτερικά και εξωτερικά σφάλματα υλικού (π.χ. βλάβη σε συσκευή).
 - Σφάλματα λογισμικού (π.χ. διαίρεση με το 0 ή προσπάθεια πρόσβασης σε απαγορευμένη περιοχή στη μνήμη).
 - Αδυναμία του Λ.Σ. να ικανοποιήσει κάποια ανάγκη μίας εφαρμογής.
 - Σε αυτές τις περιπτώσεις το Λ.Σ. πρέπει να αντιμετωπίσει το πρόβλημα που δημιουργείται με τρόπο που κατά το δυνατόν να μην επηρεάζει αρνητικά τη λειτουργία του συστήματος (π.χ. τερματισμός του προγράμματος ή επανάληψη της ενέργειας που δημιούργησε το πρόβλημα).

7

Υπηρεσίες που παρέχονται από το Λ.Σ. — 4

- Εντολές επικοινωνίας με το υλικό του συστήματος (instruction set architecture – ISA).
 - Είναι η ομάδα των εντολών που αντιλαμβάνεται ο Η/Υ, αποτελούν το σύνορο μεταξύ υλικού και λογισμικού και μπορούν να εκτελεστούν από τα προγράμματα.
- Δυαδική διασύνδεση εφαρμογών (application binary interface – ABI).
 - Είναι ένα πρότυπο για δυαδική φορητότητα μεταξύ προγραμμάτων.
- Διασύνδεση προγραμματισμού εφαρμογών (application programming interface – API).
 - Είναι ένας μηχανισμός που επιτρέπει σε προγράμματα να έχουν πρόσβαση στο υλικό και υπηρεσίες ενός συστήματος. Αυξάνει τη φορητότητα του λογισμικού μεταξύ συστημάτων που υποστηρίζουν τα ίδια API.

8

Υπηρεσίες που παρέχονται από το Λ.Σ. — 5

- Στατιστική χρήση του συστήματος.
 - Συλλογή στατιστικών στοιχείων αναφορικά με τη χρήση του συστήματος από τους χρήστες.
 - Παρακολούθηση της απόδοσης του συστήματος.
 - Η χρησιμότητα των ανωτέρω έγκειται στη συνεχή προσαρμογή του συστήματος έτσι ώστε να είναι όσο γίνεται πιο αποδοτικό και στη χρέωση των χρηστών του (όπου αυτό ισχύει).

9

Ο (διπλός) ρόλος ενός Λ.Σ.

- Ως μία εκτεταμένη (extended) ή ιδεατή (virtual) μηχανή που παρέχει μία διασύνδεση μεταξύ χρήστη και Η/Υ.
- Ως διαχειριστής των πόρων του συστήματος (resource manager).

10

Το Λ.Σ. ως λογισμικό

- Το Λ.Σ. λειτουργεί όπως ένα οποιοδήποτε άλλο είδος λογισμικού, με την έννοια ότι χρειάζεται τον επεξεργαστή για να εκτελεσθεί.
- Ως μέρος των ενεργειών του, το Λ.Σ. πολύ συχνά εκχωρεί τον έλεγχο της ΚΜΕ στα υπόλοιπα προγράμματα που εκτελούνται και αναμένει από την ΚΜΕ να του επιτρέψει να την επαναχρησιμοποιήσει.

11

Το Λ.Σ. ως διαχειριστής των πόρων του συστήματος

The diagram illustrates the computer system architecture. On the left, a box labeled 'Computer System' contains 'Memory' (with sub-sections for 'Operating System Software' and 'Programs and Data') and 'Processor' (with three dots between two 'Processor' labels). On the right, 'I/O Devices' (Printers, keyboards, digital camera, etc.) and 'Storage' (OS, Programs, Data) are shown. Arrows indicate the flow of data and control between the system components and the external devices and storage.

Figure 2.2 The Operating System as Resource Manager

12

Ιστορία και εξέλιξη των Λ.Σ.

- Τα Λ.Σ. εξελίσσονται με τον χρόνο.
 - Αναβάθμιση συσκευών και χρήση νέων συσκευών.
 - Νέες υπηρεσίες.
 - Διορθώσεις.

13

Περιεχόμενα

- Στόχοι και λειτουργίες του Λ.Σ.
- Η εξέλιξη των Λ.Σ.
- Κύρια επιτεύγματα.
- Εξελίξεις που οδήγησαν στα μοντέρνα Λ.Σ.
- Microsoft Windows.
- Συστήματα UNIX.
- Linux.
- Android.

14

Η εξέλιξη των Λ.Σ.

- Η εξέταση των διαφόρων σταδίων εξέλιξης των Λ.Σ. βοηθάει στην κατανόηση των βασικών χαρακτηριστικών τους και στην ανάγκη που οδήγησε στην ανάπτυξή τους.
- Τα στάδια αυτά συμπεριλαμβάνουν τα ακόλουθα:
 - Σειριακή επεξεργασία (serial processing).
 - Συστήματα σειριακής επεξεργασίας δέσμης (simple batch systems).
 - Συστήματα πολυπρογραμματισμού σειριακής επεξεργασίας δέσμης (multiprogrammed batch systems).
 - Συστήματα καταμερισμού χρόνου (time sharing systems).
- Η εξέλιξη των Λ.Σ. είναι σε μεγάλο βαθμό συνυφασμένη με αυτήν των Η/Υ και μπορεί να χωριστεί σε 4 γενεές.

15

Χρονική αλληλουχία

16

1^η Γενεά (1945 — 1955): Χαρακτηριστικά

- Ο Η/Υ αποτελείται από λυχνίες κενού (Cathod Ray Tubes) και κάρτες καλωδιακών συνδέσεων.
- Ουσιαστικά δεν υπάρχει Λ.Σ.
- Το «πρόγραμμα» αποτελείται από κάρτες καλωδιακών συνδέσεων (punchboards) που «έτρεχαν» στον Η/Υ (αν και όταν κάποιες από τις περίπου 20.000 λυχνίες κενού δεν ήταν καμένες).
- Προς τις αρχές του '50 γίνεται «αναβάθμιση» με τη χρήση διάτρητων καρτών (punched cards) αντί για κάρτες καλωδιακών συνδέσεων.
- Ευθύνη του χρήστη να αναλάβει ο ίδιος όλες τις λειτουργίες που σχετίζονται με την εκτέλεση κάποιου προγράμματος: φόρτωση του μεταγλωττιστή, βιβλιοθηκών, κλπ., παραγωγή συμβολικής γλώσσας, φόρτωση του συμβολομεταφραστή και παραγωγή κώδικα.

17

1^η Γενεά (1945 — 1955): Προβλήματα

- Χρονοδρομολόγηση.
 - Συνήθως ο Η/Υ δεσμεύεται από τους χρήστες σε χρονικά διαστήματα της 0.5 ώρας.
 - Ένας χρήστης δεσμεύει το σύστημα για μισή ώρα αλλά το χρησιμοποιεί μόνο 20 λεπτά. Κατ' επέκταση το σύστημα είναι ανενεργό για 10 λεπτά.
 - Ο χρήστης δεν καταφέρνει να τελειώσει σε μισή ώρα και αναγκάζεται να φύγει, χωρίς να έχει ολοκληρώσει τη δουλειά του.
- Χρόνος προετοιμασίας του συστήματος.
 - Ο χρήστης είναι υποχρεωμένος να αναλάβει ο ίδιος όλες τις διαδικασίες εκτέλεσης του προγράμματος που συμπεριλαμβάνουν τη μεταφορά ταινιών και τη φόρτωσή τους, διάβασμα διάτρητων καρτελών, κλπ.
 - Αν συμβεί κάποιο σφάλμα, ο άτυχος χρήστης πρέπει να ξεκινήσει πάλι από την αρχή.
- Συμπερασματικά, γίνεται μεγάλη σπατάλη χρόνου που οδηγεί σε σημαντική υποχρησιμοποίηση του Η/Υ, που εκείνη την εποχή ήταν πολύ ακριβός.

18

2^η Γενιά (1955 — 1965): Χαρακτηριστικά

- Ο Η/Υ αποτελείται από κρυσταλλοτριόδους (transistors) που είναι πιο αξιόπιστα από τις λυχνίες. Αυτό έχει θετικό αντίκτυπο και στην ανάπτυξη του λογισμικού.
- Αναπτύσσεται η έννοια της εργασίας (job), αποτελούμενη από ένα ή περισσότερα προγράμματα γραμμένα σε κάποια γλώσσα προγραμματισμού μαζί με τα δεδομένα τους. Κάθε εργασία διαβάζεται από τον Η/Υ με χρήση διάτρητων καρτών.
- Δημιουργείται η έννοια της σειριακής επεξεργασίας (serial processing): μόλις τελειώνει η εκτέλεση μίας εργασίας αρχίζει αμέσως να εκτελείται η επόμενη.
- Για την πιο αποδοτική χρήση του Η/Υ, ο χρήστης παύει να έχει άμεσο έλεγχο του συστήματος αλλά παραδίδει τις κάρτες με το πρόγραμμά του σε κάποιον χειριστή ο οποίος τις φορτώνει στον Η/Υ.

19

Παρακολουθητής

- Η σειριακή επεξεργασία και η εκτέλεση ενός προγράμματος μετά το άλλο, γίνεται δυνατή με τη χρήση ενός λογισμικού που λέγεται παρακολουθητής (monitor), μέρος του οποίου βρίσκεται πάντα στη μνήμη.
- Ο παρακολουθητής διαβάζει μία εργασία η οποία φορτώνεται στην περιοχή χρηστών της μνήμης και της δίνει τον έλεγχο της ΚΜΕ για να εκτελεσθεί. Η κάθε εργασία διαμορφώνεται με τέτοιο τρόπο έτσι ώστε μόλις ολοκληρώσει την εκτέλεσή της, ο έλεγχος επιστρέφει στον παρακολουθητή ο οποίος φορτώνει την επόμενη εργασία, κ.ο.κ.
- Τα αποτελέσματα της εκτέλεσης της κάθε εργασίας στέλνονται σε κάποια αυθεντική εξόδου για να τα παραλάβει ο χρήστης.

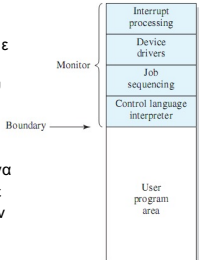


Figure 2.3 Memory Layout for a Resident Monitor

20

Γλώσσες Ελέγχου Εργασιών

- Οι γλώσσες ελέγχου εργασιών (job control languages) παρέχουν στον παρακολουθητή σημαντικές οδηγίες για την εκτέλεση της κάθε εργασίας, όπως:
 - Ποιον μεταφραστή να χρησιμοποιήσει.
 - Ποια δεδομένα να χρησιμοποιήσει.
- Πριν την εκτέλεση ενός προγράμματος, ο παρακολουθητής διαβάζει τις κάρτες που αντιστοιχούν στις οδηγίες αυτές και φορτώνει αμέσως τα βοηθητικά προγράμματα που χρειάζονται (μεταφραστές, συμβολομεταφραστές, κλπ.), ελαττώνοντας έτσι σημαντικά τη σπατάλη χρόνου μεταξύ της εκτέλεσης δύο εργασιών.

21

Επιπτώσεις από τη χρήση του παρακολουθητή

- Το προαναφερθέν μοντέλο εκτέλεσης εργασιών, για να λειτουργήσει σωστά, πρέπει να υποστηρίζει τα ακόλουθα:
 - Προστασία μνήμης: Ο χώρος της μνήμης που χρησιμοποιείται από τον παρακολουθητή δεν πρέπει να είναι προσπελάσιμος στο πρόγραμμα του χρήστη που εκτελείται.
 - Χρονικό μετρητή: Ένα πρόγραμμα χρήστη δεν πρέπει να εκτελείται επ' άοριστον.
 - Προνομίους εντολές (privileged instructions): Όταν ένα πρόγραμμα χρήστη ζητήσει την εκτέλεση μίας τέτοιας εντολής, τότε ο έλεγχος πρέπει να επιστρέψει (πιθανόν προσωρινά) στον παρακολουθητή (π.χ. εντολές εισόδου/εξόδου).
 - Διακόπτες (interrupts). Επιτρέπει την πιο εύκολη και αποδοτική μετάπτωσηση ελέγχου μεταξύ του παρακολουθητή και του εκτελούμενου προγράμματος.

22

Τρόποι λειτουργίας ενός Η/Υ

- Σε κατάσταση χρήστη (user mode), όπου το πρόγραμμα εκτελείται στο χώρο μνήμης του χρήστη και η υπόλοιπη μνήμη δεν είναι προσπελάσιμη σε αυτό. Επιπλέον, κάποιες εντολές δεν μπορούν να εκτελεστούν.
- Σε κατάσταση πυρήνα (kernel mode), όπου εκτελείται ο παρακολουθητής ο οποίος έχει πρόσβαση σε όλη τη διαθέσιμη μνήμη του συστήματος και μπορεί να εκτελέσει οποιαδήποτε εντολή.
- Για όλα τα ανωτέρω, ο παρακολουθητής θεωρείται ο προπομπός του Λ.Σ., όπως το αντιλαμβανόμαστε σήμερα.

23

2^η Γενιά (1955 — 1965): Προβλήματα

- Ακόμα και με την εισαγωγή του παρακολουθητή και της σειριακής επεξεργασίας, η ΚΜΕ, η πιο σημαντική μονάδα ενός Η/Υ, υποχρησιμοποιείται σε απaráδεκτα χαμηλό ρυθμό.
- Το πρόβλημα βρίσκεται στο ότι οι συσκευές Ε/Ε είναι κατά κανόνα πολύ πιο αργές από την ΚΜΕ. Π.χ.

Διάβασμα μίας εγγραφής από κάποιο αρχείο:	15 μs
Εκτέλεση 100 εντολών:	1 μs
Γράψιμο μίας εγγραφής πίσω στο αρχείο:	15 μs
Σύνολο χρόνου:	31 μs

Ποσοστιαία χρήση ΚΜΕ: $1/31 = 0,032 = 3,2\%$

24

3η Γενιά (1965 — 1980): Χαρακτηριστικά

- Ο Η/Υ αποτελείται από ολοκληρωμένα κυκλώματα (large scale integration), που βελτιώνουν την ταχύτητα και αυξάνουν τη χωρητικότητα μνήμης σε έναν Η/Υ.
- Αναπτύσσονται νέες τεχνικές, όπως:
 - Πολυπρογραμματισμός (multiprogramming): Η ταυτόχρονη ύπαρξη στην μνήμη περισσότερων της μίας εργασιών και η εναλλακτική εκτέλεσή τους στην ΚΜΕ.
 - Χρονοδρομολόγηση (scheduling): Πολιτικές με βάση τις οποίες καθορίζεται με ποια σειρά και για πόσο χρονικό διάστημα θα εκτελείται η κάθε εργασία.
 - Διαχείριση μνήμης (memory management): Πολιτικές με βάση τις οποίες καθορίζεται πως θα μοιραστεί η υπάρχουσα μνήμη στις εργασίες που είναι φορτωμένες στο σύστημα.
 - Καταμερισμός χρόνου (time sharing): Η υποστήριξη ταυτόχρονης σύνδεσης σε ένα Η/Υ από πολλούς χρήστες μέσω τερματικών σταθμών.

25

Μονοπρογραμματισμός

- Στον μονοπρογραμματισμό (uniprogramming), η ΚΜΕ μόλις συναντήσει μία εντολή Ε/Ε είναι υποχρεωμένη να περιμένει σε ανενεργή κατάσταση μέχρις ότου ολοκληρώσει την εκτέλεσή της αυτή η εντολή, για να συνεχίσει την εκτέλεση των επόμενων (υπολογιστικών) εντολών του προγράμματος.

(a) Uniprogramming

26

Πολυπρογραμματισμός — 1

- Αν όμως στη μνήμη υπήρχαν 2 προγράμματα, η ΚΜΕ θα ήταν ανενεργή για μικρότερο χρονικό διάστημα.

(b) Multiprogramming with two programs

27

Πολυπρογραμματισμός — 2

- Στη γενική περίπτωση, όσα περισσότερα προγράμματα υπάρχουν ταυτόχρονα στη μνήμη, τόσο μικρότερο αναμένεται να είναι το χρονικό διάστημα στο οποίο η ΚΜΕ θα βρίσκεται σε ανενεργή κατάσταση.

(c) Multiprogramming with three programs

28

Παράδειγμα

Table 2.1 Sample Program Execution Attributes

	JOB1	JOB2	JOB3
Type of job	Heavy compute	Heavy I/O	Heavy I/O
Duration	5 min	15 min	10 min
Memory required	50 M	100 M	75 M
Need disk?	No	No	Yes
Need terminal?	No	Yes	No
Need printer?	No	No	Yes

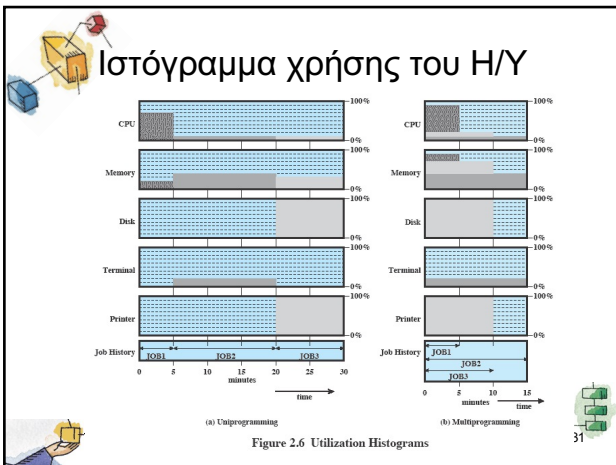
29

Τα πλεονεκτήματα του πολυπρογραμματισμού

Table 2.2 Effects of Multiprogramming on Resource Utilization

	Uniprogramming	Multiprogramming
Processor use	20%	40%
Memory use	33%	67%
Disk use	33%	67%
Printer use	33%	67%
Elapsed time	30 min	15 min
Throughput	6 jobs/hr	12 jobs/hr
Mean response time	18 min	10 min

30



31

Συστήματα Καταμερισμού Χρόνου

- Σε πολλές περιπτώσεις, ο χρήστης πρέπει να έχει άμεση και διαλογική (interactive) πρόσβαση στο σύστημα.
- Σε αυτές τις περιπτώσεις οι χρήστες χρησιμοποιούν το σύστημα μέσω της χρήσης τερματικών σταθμών.
- Η ΚΜΕ πρέπει να διαμοιράζει το χρόνο της, εξυπηρετώντας ταυτόχρονα τις ανάγκες όλων των χρηστών.

32

Σειριακή Επεξεργασία Δέσμης vs. Καταμερισμός Χρόνου

Table 2.3 Batch Multiprogramming versus Time Sharing

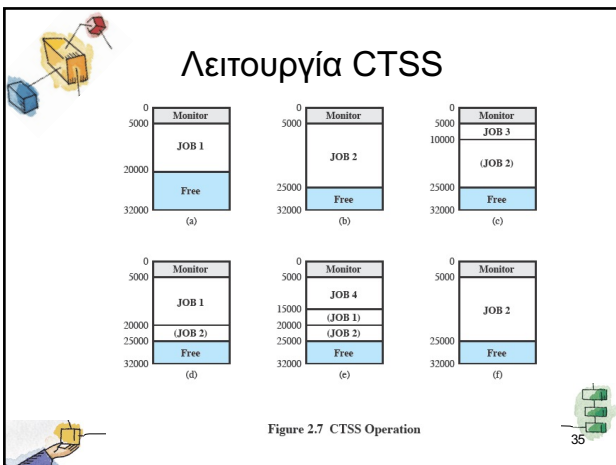
	Batch Multiprogramming	Time Sharing
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

33

Το πρώτο παράδειγμα Λ.Σ. καταμερισμού χρόνου: CTSS

- [Compatible Time-Sharing System](#) (CTSS)
 - Αναπτύχθηκε στο MIT στα πλαίσια του προγράμματος MAC.
- Το σύστημα είχε 32.000 λέξεις μνήμης, από τις οποίες ο παρακολουθητής χρησιμοποίησε τις 5.000.
- Ένα πρόγραμμα φορτωνόταν στη μνήμη ξεκινώντας πάντα από τη θέση 5,000.
- Κάθε 0,2 δευτερόλεπτα ένας διακόπτης μετέφερε τον έλεγχο του συστήματος στον παρακολουθητή ο οποίος είχε τη δυνατότητα να δώσει την ΚΜΕ σε ένα άλλο πρόγραμμα για να εκτελεσθεί.
- Αν υπήρχε ανάγκη, το πρόγραμμα από το οποίο έφευγε ο έλεγχος χρήσης της ΚΜΕ αποθηκευόταν στο δίσκο, για να φορτωθεί ξανά σε κάποιο μελλοντικό στάδιο.

34




35

3^η Γενιά (1965 — 1980): Προβλήματα


- Τα βασικά χαρακτηριστικά της 3^{ης} γενιάς αποτελούν ουσιαστικά και προβλήματα τα οποία πρέπει να αντιμετωπισθούν.
- Η ταυτόχρονη ύπαρξη πολλών προγραμμάτων στη μνήμη, δημιουργεί την ανάγκη διαφύλαξης των δεδομένων και πληροφοριών ενός προγράμματος από τα υπόλοιπα (π.χ. ένα πρόγραμμα να τροποποιεί τα δεδομένα κάποιου άλλου προγράμματος).
- Με την ταυτόχρονη πρόσβαση στο σύστημα από πολλούς χρήστες, τα αρχεία ενός χρήστη πρέπει να προστατευθούν από μη εξουσιοδοτημένη πρόσβαση σε αυτά από άλλους χρήστες.
- Ο ανταγωνισμός από πολλαπλά προγράμματα και χρήστες για πρόσβαση σε κοινούς πόρους του συστήματος πρέπει να επιλυθεί με τρόπο δίκαιο και ικανοποιητικό για όλους τους εμπλεκόμενους.

36




4η Γενιά (1980 — τώρα): Χαρακτηριστικά

- Ο Η/Υ αποτελείται από κυκλώματα μεγάλης και πολύ μεγάλης ολοκλήρωσης (LSI, VLSI) τα οποία μείωσαν δραστικά το κόστος του με αποτέλεσμα την ανάπτυξη προσωπικών Η/Υ.
- Ανάπτυξη Λ.Σ. ενός χρήστη με έμφαση στη φιλικότητα προς τον χρήστη (π.χ. DOS, OS/2, Windows, OS X).
- Ανάπτυξη ισχυρών σταθμών εργασίας (workstations) και δικτύων που οδήγησαν στην ανάπτυξη λειτουργικών συστημάτων για δίκτυα και κατακεντρωμένων (distributed) λειτουργικών συστημάτων.




37




Περιεχόμενα

- Στόχοι και λειτουργίες του Λ.Σ.
- Η εξέλιξη των Λ.Σ.
- **Κύρια επιτεύγματα.**
- Εξελίξεις που οδήγησαν στα μοντέρνα Λ.Σ.
- Microsoft Windows.
- Συστήματα UNIX.
- Linux.
- Android.




38




Βαθμός δυσκολίας ανάπτυξης ενός Λ.Σ.

- Το Λ.Σ. είναι ένα από τα πιο πολύπλοκα είδη λογισμικού που αναπτύσσονται.
- Μερικά σχετικά στατιστικά στοιχεία:
 - Το Multics ανακοινώθηκε το 1963 και λειτούργησε το 1969.
 - Το OS 360 όταν πρωτοβγήκε στην αγορά είχε 1000 λάθη.
 - Το Windows NT υπέφερε επί 7 χρόνια με προβλήματα.
 - Το Windows Vista ουσιαστικά ποτέ δεν καθιερώθηκε και αντικαταστάθηκε πολύ γρήγορα από το Windows 7.
 - Ένα μικρό Λ.Σ. έχει μέγεθος 100K και ένα μεγάλο 10M γραμμών κώδικα.
 - Η δημιουργία ενός Λ.Σ. χρειάζεται μεταξύ 100-1000 ανθρωποετών (person years).




39




Κύρια επιτεύγματα

- Η έννοια της διεργασίας (process).
- Διαχείριση μνήμης.
- Προστασία και ασφάλεια των πληροφοριών σε ένα σύστημα.
- Χρονοδρομολόγηση.
- Διαχείριση των πόρων του συστήματος.




40




Διεργασία

- Η οντότητα εκείνη που εκτελείται από τον επεξεργαστή. Ανά πάσα στιγμή στο σύστημα υπάρχουν πολλές διεργασίες, μία από τις οποίες εκτελείται ενώ οι υπόλοιπες περιμένουν τη σειρά τους.
- Μια διεργασία μπορεί να δημιουργήσει άλλες θυγατρικές διεργασίες και να αναμένει (ή όχι) τον τερματισμό τους πριν συνεχίσει τη δική της εκτέλεση.
- Μια διεργασία μπορεί να είναι ένα πρόγραμμα, ένα κομμάτι προγράμματος ή οτιδήποτε εκτελείται από την ΚΜΕ ως μία σειρά εντολών.




41



Πηγές λαθών με τη χρήση διεργασιών

- Η χρήση διεργασιών, αν και οδηγεί στην ανάπτυξη ευέλικτων Λ.Σ., συχνά αποτελεί αιτία για τη δημιουργία λαθών, τα οποία κατά κανόνα είναι δύσκολο να εντοπισθούν αλλά και να βρεθούν οι λόγοι που τα δημιούργησαν.
- Τα λάθη προέρχονται κυρίως από:
 - Μη καλό συγχρονισμό μεταξύ των διεργασιών που εμπλέκονται σε κάποια κοινή ενέργεια.
 - Μη σωστή πρόσβαση από πολλές διεργασίες σε κοινούς πόρους.
 - Μη προκαθορισμένη συμπεριφορά του συστήματος.
 - Δημιουργία αδιέξοδου όπου ένας αριθμός από διεργασίες που εμπλέκονται σε κάποια κοινή ενέργεια δεν μπορούν να συνεχίσουν την εκτέλεσή τους.



42

Συστατικά μίας διεργασίας

- Μια διεργασία αποτελείται από:
 - Ένα εκτελέσιμο κομμάτι προγράμματος.
 - Τα δεδομένα που σχετίζονται με την εκτέλεση του κώδικα της διεργασίας.
 - Το περιβάλλον στο οποίο εκτελείται η διεργασία και το οποίο έχει όλες τις πληροφορίες που χρειάζεται το Λ.Σ. για να διαχειριστεί τη διεργασία και το οποίο περιλαμβάνει τις τιμές των καταχωρητών, τον ιδιοκτήτη και προτεραιότητα της διεργασίας, κλπ.

43

Διαχείριση διεργασιών

The diagram illustrates the typical implementation of a process. It shows a vertical stack of memory blocks. At the top is the 'Main Memory'. Below it, a 'Process List' contains entries for 'Process i', 'Process j', 'Process A', and 'Process B'. Each process entry points to a specific memory region containing 'Control', 'Data', and 'Program (code)'. To the right, 'Processor Registers' are shown, including 'PC', 'Base Register', and 'Other registers'. Arrows indicate the mapping between the process list and the memory blocks, and between the registers and the process memory.

Figure 2.8 Typical Process Implementation

44

Διαχείριση μνήμης

- Το Λ.Σ. έχει 5 σημεία ευθύνης σε αυτόν τον τομέα:
 - Απομόνωση του χώρου μνήμης μίας διεργασίας από αυτούς των άλλων διεργασιών.
 - Αυτόματη διαμοίραση της μνήμης στις διεργασίες και διαχείρισή της.
 - Υποστήριξη σε αρθρωτό (modular) προγραμματισμό.
 - Έλεγχος και προστασία κατά την πρόσβαση των διεργασιών στη μνήμη του συστήματος.
 - Μακρόχρονη αποθήκευση πληροφοριών.

45

Ιδεατή Μνήμη

- Η τεχνική της ιδεατής μνήμης (virtual memory) επιτρέπει στα προγράμματα να βλέπουν τη μνήμη από μία λογική οπτική γωνία, χωρίς να χρειάζεται να γνωρίζουν πόση πραγματική μνήμη είναι στην ουσία διαθέσιμη.

46

Σελιδοποίηση

- Η τεχνική της σελιδοποίησης (paging) επιτρέπει σε μία διεργασία να αποτελείται από ένα αριθμό από ιδίου μεγέθους κομμάτια που λέγονται σελίδες.
- Κάθε ιδεατή διεύθυνση μνήμης αποτελείται από ένα αριθμό σελίδας και μία μετατόπιση μέσα στη σελίδα.
- Η κάθε σελίδα μπορεί να βρίσκεται οπουδήποτε στη μνήμη.

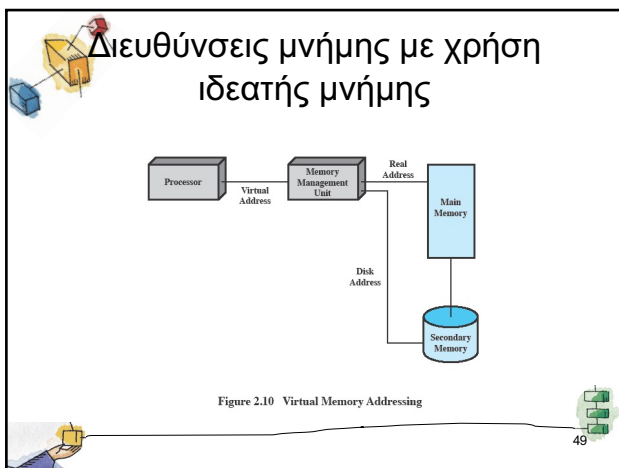
47

Παράδειγμα ιδεατής μνήμης

The diagram shows two memory structures. On the left, 'Main Memory' is a grid of frames. Some frames are labeled with addresses: A.1, A.2, A.5, B.0, B.1, B.2, B.3, A.7, A.8, B.5, B.6. On the right, 'Disk' is a cylinder with pages numbered 0 to 10. Pages 0-6 are labeled 'User program A' and pages 7-10 are labeled 'User program B'. Arrows indicate the mapping between the memory frames and the disk pages.

Figure 2.9 Virtual Memory Concepts

48



49

Διαφύλαξη και προστασία των πληροφοριών

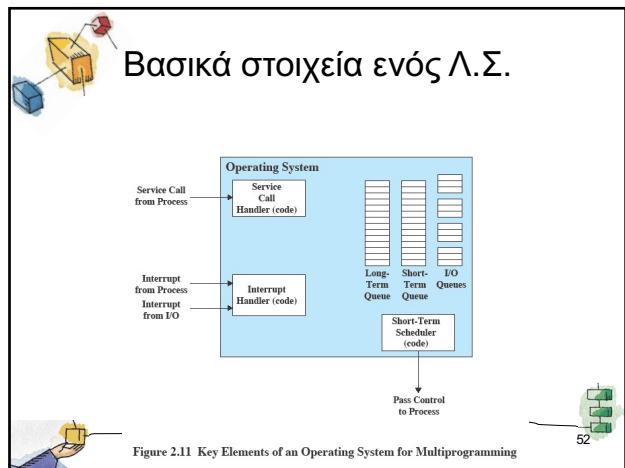
- Το πρόβλημα σχετίζεται με τον έλεγχο πρόσβασης σε ένα σύστημα και των πληροφοριών που είναι αποθηκευμένες σε αυτό.
- Τα βασικά θέματα εδώ μπορούν να χωρισθούν σε 3 βασικές κατηγορίες:
 - Διαθεσιμότητα των πληροφοριών.
 - Αποφυγή πρόσβασης στις πληροφορίες και πιθανόν αλλοίωσής τους από μη εξουσιοδοτημένα άτομα.
 - Επιβεβαίωση της ταυτότητας και των στοιχείων του συστήματος.

50

Χρονοδρομολόγηση και διαχείριση των πόρων

- Όσον το δυνατόν πιο αποτελεσματική διαχείριση των πόρων του συστήματος με βάση τα εξής κριτήρια:
 - Δικαιοσύνη, δηλαδή όλες οι διεργασίες κάποια στιγμή να εξυπηρετηθούν.
 - Αναγνώριση τυχόν ιδιαιτεροτήτων κάποιων διεργασιών, όπως σύντομος χρόνος χρήσης της ΚΜΕ, μικρή ανάγκη σε μνήμη, προτεραιότητα, κλπ.
 - Αποδοτικότητα του συστήματος, δηλαδή μεγιστοποίηση του ωφέλιμου έργου που επιτελείται και αριθμού χρηστών που εξυπηρετούνται σε ένα χρονικό διάστημα και ταυτόχρονα ελαχιστοποίηση του χρόνου εξυπηρέτησης του κάθε χρήστη ξεχωριστά.

51



52

Στρωματοποιημένα Λ.Σ.

- Το Λ.Σ. αποτελείται από μία ιεραρχία επιπέδων, το καθένα από τα οποία επικοινωνεί κατ' εξοχήν μόνο με το αμέσως προηγούμενο και επόμενο.
- Το κάθε επίπεδο εκτελεί μόνο ένα υποσύνολο των λειτουργιών ενός Λ.Σ.
- Το κάθε επίπεδο βασίζεται στο αμέσως επόμενο για την εκτέλεση πιο στοιχειωδών εντολών.
- Επειδή ιδανικά αλλαγές σε ένα επίπεδο δεν επηρεάζουν άλλα επίπεδα, η διαχείριση ενός Λ.Σ. και η αντιμετώπιση των προβλημάτων που προκύπτουν αντιμετωπίζεται πιο εύκολα.
- Ο αριθμός των επιπέδων μπορεί να ποικίλει από ένα Λ.Σ. σε άλλο αλλά οι Denning και Brown περιγράφουν μία γενικευμένη δομή που αποτελείται από 13 επίπεδα.

53


Ιεραρχία επιπέδων ενός Λ.Σ.

Table 2.4 Operating System Design Hierarchy

Level	Name	Objects	Example Operations
13	Shell	User programming environment	Statements in shell language
12	User processes	User processes	Quit, kill, suspend, resume
11	Directories	Directories	Create, destroy, attach, detach, search, list
10	Devices	External devices, such as printers, displays, and keyboards	Open, close, read, write
9	File system	Files	Create, destroy, open, close, read, write
8	Communications	Pipes	Create, destroy, open, close, read, write
7	Virtual memory	Segments, pages	Read, write, fetch
6	Local secondary store	Blocks of data, device channels	Read, write, allocate, free
5	Primitive processes	Primitive processes, semaphores, ready list	Suspend, resume, wait, signal
4	Interrupts	Interrupt-handling programs	Invoke, mask, unmask, retry
3	Procedures	Procedures, call stack, display	Mark stack, call, return
2	Instruction set	Evaluation stack, microprogram interpreter, scalar and array data	Load, store, add, subtract, branch
1	Electronic circuits	Registers, gates, buses, etc.	Clear, transfer, activate, complement


Gray shaded area represents hardware.

54




Περιοχές ευθύνης του Λ.Σ.

- Διαχείριση διεργασιών
- Διαχείριση κύριας μνήμης
- Διαχείριση περιφερειακής μνήμης
- Διαχείριση συσκευών E/E
- Τα ανωτέρω αποτελούν επίσης τις βασικές ενότητες ενός μαθήματος εισαγωγής σε Λ.Σ.




55




Περιεχόμενα

- Στόχοι και λειτουργίες του Λ.Σ.
- Η εξέλιξη των Λ.Σ.
- Κύρια επιτεύγματα.
- – Εξελίξεις που οδήγησαν στα μοντέρνα Λ.Σ.
- Microsoft Windows.
- Συστήματα UNIX.
- Linux.
- Android.




56




Διαφορετικές αρχιτεκτονικές προσεγγίσεις

- Αυτές περιλαμβάνουν τις ακόλουθες:
 - Μικροπυρήνες (microkernel).
 - Πολυνημάτωση (multithreading).
 - Συμμετρική πολυεπεξεργασία (symmetric multiprocessing).
 - Κατανεμημένα Λ.Σ.
 - Αντικειμενοστρεφής σχεδιασμός.




57




Αρχιτεκτονικές Μικροπυρήνα

- Στα πρώτα Λ.Σ. όλες σχεδόν οι λειτουργίες τους βρισκόταν στον πυρήνα ο οποίος υλοποιούταν ως μία διεργασία με όλες τις εμπλεκόμενες οντότητες να διαμοιράζονται ένα κοινό χώρο μνήμης.
- Με τη χρήση μικροπυρήνα μόνο μερικές βασικές λειτουργίες βρίσκονται σε αυτόν:
 - Χώρος διευθύνσεων.
 - Επικοινωνία μεταξύ των διεργασιών.
 - Βασικοί μηχανισμοί χρονοδρομολόγησης.
- Οι υπόλοιπες λειτουργίες παρέχονται από διεργασίες, συχνά ονομαζόμενες εξυπηρετητές, οι οποίες εκτελούνται σε κατάσταση χρήστη, προσαρμόζονται εύκολα σε συγκεκριμένες ανάγκες και ο μικροπυρήνας τις αντιμετωπίζει ως να είναι κοινές διεργασίες.
- Με αυτόν τον τρόπο τα Λ.Σ. καθίστανται πιο εύλικτα και η υλοποίησή τους είναι πιο εύκολη.




58




Πολυνημάτωση

- Μια διεργασία διαιρείται σε υποδιεργασίες (ή νήματα) οι οποίες εκτελούνται ταυτόχρονα.
- Η υποδιεργασία:
 - Είναι μία σχετικά μικρή και αυτόνομη εκτελούμενη οντότητα με το δικό της περιβάλλον (καταχωρητές, στοίβες, μνήμη, κλπ.).
 - Εκτελείται σειριακά αλλά η εκτέλεσή της μπορεί να ανασταλεί αν η ΚΜΕ πρέπει να εκτελέσει μία άλλη υποδιεργασία.
- Με τη διάσπαση των διεργασιών που αποτελούν μία εφαρμογή σε υποδιεργασίες, ο προγραμματιστής έχει μεγαλύτερη ευελιξία στην εκτέλεση της εφαρμογής και υποστήριξη στη δημιουργία αρθρωτών προγραμμάτων.
- Η πολυνημάτωση είναι χρήσιμη σε εφαρμογές που εκτελούν ένα αριθμό από εργασίες οι οποίες είναι ανεξάρτητες μεταξύ τους.




59



Συμμετρική Πολυεπεξεργασία


- Ένα τέτοιο σύστημα αποτελείται από πολλούς επεξεργαστές οι οποίοι διαμοιράζονται την ίδια μνήμη και μονάδες E/E και εκτελούν τις ίδιες λειτουργίες (γι' αυτό και ο όρος συμμετρική).
- Σε ένα τέτοιο σύστημα, το Λ.Σ. χρησιμοποιεί ταυτόχρονα όλους τους επεξεργαστές για την παράλληλη εκτέλεση των διεργασιών.



60

Πλεονεκτήματα της συμμετρικής πολυεπεξεργασίας

- Μεγαλύτερη απόδοση από τα συμβατικά συστήματα, λόγω της παράλληλης εκτέλεσης πολλών εργασιών που μειώνει το συνολικό χρόνο εκτέλεσης μίας εφαρμογής.
- Αν προκύψει πρόβλημα με κάποιους από τους επεξεργαστές, το σύστημα είναι σε θέση να συνεχίσει να λειτουργεί με τους υπόλοιπους.
- Το σύστημα μπορεί να επεκταθεί εύκολα με επιπρόσθετους επεξεργαστές.
- Τα συστήματα αυτά είναι κλιμακωτά και υπάρχουν διάφορα μοντέλα σε σχέση με απόδοση ή κόστος.



61

61

Πολυπρογραμματισμός vs. Πολυεπεξεργασία

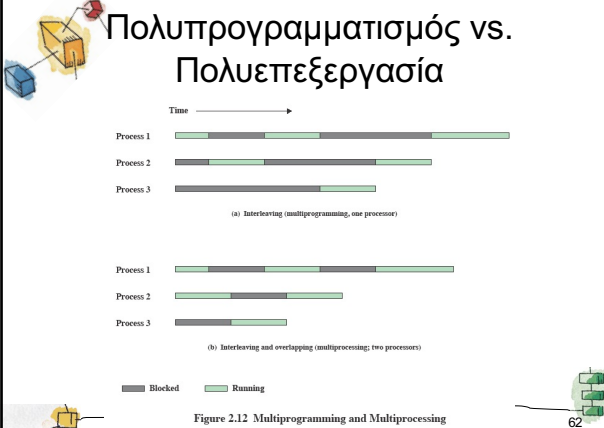



Figure 2.12 Multiprogramming and Multiprocessing




62

62

Κατανεμημένα Λ.Σ.

- Παρέχουν την ψευδαίσθηση ότι υπάρχει ένα κοινός χώρος μνήμης (κύριας ή περιφερειακής), παρόλο που σε φυσικό επίπεδο η μνήμη είναι κατανεμημένη μεταξύ διαφορετικών Η/Υ.
- Το ίδιο συμβαίνει και με άλλες οντότητες στο σύστημα, όπως τα αρχεία, συσκευές, κλπ.




63

63

Αντικειμενοστρεφής σχεδιασμός

- Επιτρέπει την αρθρωτή υλοποίηση των επί μέρους συστατικών τμημάτων ενός Λ.Σ. και βοηθάει στην ανάπτυξη κατανεμημένων εργαλείων και Λ.Σ.




64

64

Περιεχόμενα

- Στόχοι και λειτουργίες του Λ.Σ.
- Η εξέλιξη των Λ.Σ.
- Κύρια επιτεύγματα.
- Εξελίξεις που οδήγησαν στα μοντέρνα Λ.Σ.
- **Microsoft Windows.**
- Συστήματα UNIX.
- Linux.
- Android.




65

65

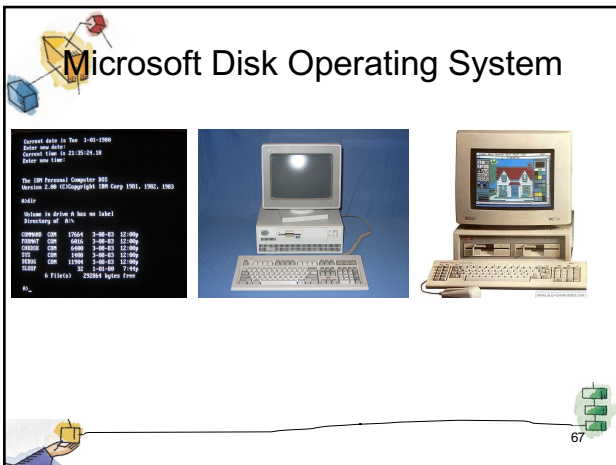
Ιστορική εξέλιξη των Windows

- **MS-DOS 1.0, 1981**
 - 4000 γραμμές κώδικα assembly
 - Έτρεχε σε 8 Kbytes μνήμη
 - Χρησιμοποιούσε επεξεργαστή Intel 8086
- **Windows 3.0, 1990**
 - 16-bit
 - Υποστήριξη παραθύρων και GUI
 - Υλοποίηση πάνω από το MS-DOS
- **Windows 95**
 - 32-bit
 - Η βάση για την ανάπτυξη των Windows 98 and Windows Me
- **Windows NT (3.1), 1993**
 - 32-bit με δυνατότητα υποστήριξης εφαρμογών σε DOS, OS/2 και προηγούμενων εκδόσεων των Windows
- **Windows 2000**
 - Υποστήριξη κατανεμημένης επεξεργασίας
 - Active Directory
 - Δυνατότητα plug-and-play
- **Windows XP, 2001**
 - Βασισμένο πάνω στα NT αντί για το MS-DOS
- **Windows Vista, 2007**
- **Windows Server, 2008**
- **Windows 7, 2009 και Windows Server 2008 R2**
- **Windows 8, 2012 and Windows 8.1, 2013**
- **Windows 10, 2015**
- **Windows Azure**
 - Υποστήριξη νεφελώδους υπολογισμού (cloud-computing)



66

66



67

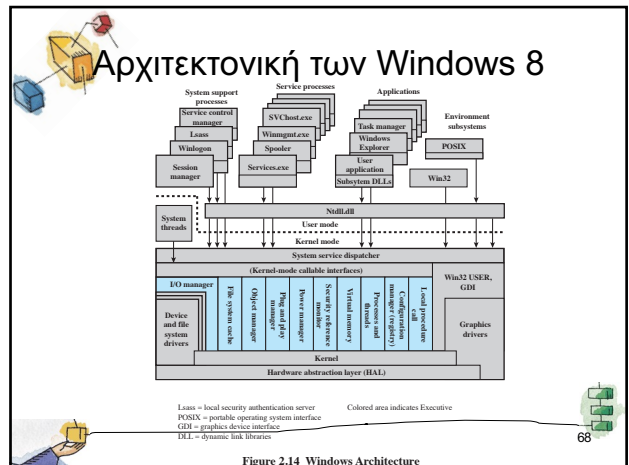


Figure 2.14 Windows Architecture

68

Πολυδιεργασία (multitasking) ενός χρήστη

- Από τα Windows 2000 και μετά, τα Windows αναπτύχθηκαν για να μπορούν να εκμεταλλεύονται τις μοντέρνες αρχιτεκτονικές Η/Υ που χρησιμοποιούν 32-bit και 64-bit επεξεργαστές.
- Αν και παραμένουν σχεδιασμένα για χρήση από ένα χρήστη, εν τούτους υποστηρίζουν πολυδιεργασία, η οποία είναι αναγκαία λόγω της αυξανόμενης απόδοσης των επεξεργαστών και της χρήσης ιδεατής μνήμης.

69

Το μοντέλο της Πελατειακής Δομής (Client/Server)

- Τα Windows χρησιμοποιούν το μοντέλο της πελατειακής δομής για την υλοποίηση των υπηρεσιών που προσφέρουν.
- Οι υπηρεσίες σε επίπεδο χρήστη υλοποιούνται ως διεργασίες που επικοινωνούν μεταξύ τους με το πρωτόκολλο RPC.
- Όταν μία εφαρμογή (client) χρειάζεται μία υπηρεσία, ένα μήνυμα στέλνεται μέσω του Executive σε κάποιον εξυπηρετητή (server) ο οποίος εκτελεί τις απαραίτητες εργασίες και επιστρέφει το ζητούμενο αποτέλεσμα στην εφαρμογή, πάλι μέσω του Executive.

70

Αντικειμενοστρεφής σχεδιασμός των Windows

- Τα Windows χρησιμοποιούν σε μεγάλο βαθμό την αντικειμενοστρεφή σχεδίαση, συμπεριλαμβανομένων των αρχών της κληρονομικότητας, πολυμορφισμού, τάξεων και ενθυλάκωσης.

71

Καινοτομίες και βελτιώσεις στα Windows 7 & 8 & 10

- Είναι κτισμένα σε επίπεδα τα οποία μπορούν να ελεγχθούν ξεχωριστά το ένα από το άλλο.
- Χρειάζονται λιγότερη μνήμη για την εκτέλεσή τους.
- Μεγαλύτερη αξιοπιστία, καλύτερη ασφάλεια και λιγότερη χρήση ενέργειας.
- Καλύτερη διαχείριση των διεργασιών.

72

Περιεχόμενα

- Στόχοι και λειτουργίες του Λ.Σ.
- Η εξέλιξη των Λ.Σ.
- Κύρια επιτεύγματα.
- Εξελίξεις που οδήγησαν στα μοντέρνα Λ.Σ.
- Microsoft Windows.
- - Συστήματα UNIX.
- Linux.
- Android.

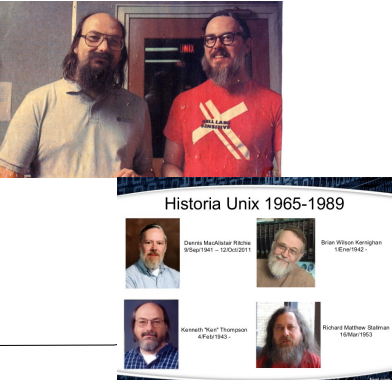
73

Ιστορική εξέλιξη του UNIX





- Σχεδιάστηκε στα εργαστήρια Bell και έτρεξε για πρώτη φορά το 1970 σε υπολογιστή PDP-7.
- Χρησιμοποίησε πολλές από τις ιδέες του Multics.
- Με την υποστήριξή του από τον υπολογιστή PDP-11 έδειξε ότι μπορεί να χρησιμοποιηθεί από διαφορετικούς Η/Υ.
- Κατόπιν υλοποιήθηκε σε υψηλού επιπέδου γλώσσα (C), κάτι πρωτοποριακό για την εποχή.
- Δημοσιεύτηκε σε επιστημονικό περιοδικό το 1974.
- Η πρώτη ευρέως διαδεδομένη έκδοσή του, εκτός των εργαστηρίων Bell, ήταν η 6^η το 1976.
- Η 7^η έκδοση του 1978 θεωρείται ο πρόγονος των μοντέρνων συστημάτων Unix.
- Δύο από τις πιο σημαντικές παραλλαγές του είναι η ομάδα AT&T και BSD του πανεπιστημίου του Berkeley.

74

C & Unix

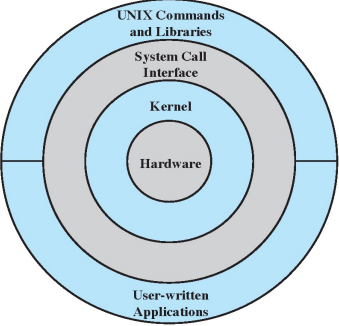


Historia Unix 1965-1989

 Dennis MacAuliffe Ritchie 8Sep1941 - 12Oct2011	 Brian Wilson Kernighan 15Nov1942 -
 Kenneth Thompson 4Feb1943 -	 Richard Matthew Stallman 18Mar1953 -

75

Το μοντέλο του UNIX



The diagram shows a layered model of UNIX. At the center is 'Hardware'. Surrounding it is the 'Kernel'. Above the kernel is the 'System Call Interface'. The outermost layer consists of 'UNIX Commands and Libraries' and 'User-written Applications'.

76

Παραδοσιακός πυρήνας UNIX

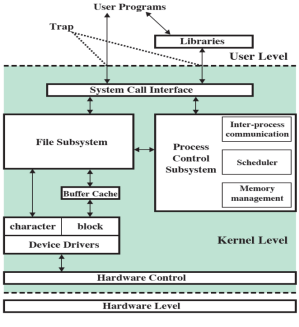


Figure 2.15 Traditional UNIX Kernel

The diagram illustrates the traditional UNIX kernel architecture. It is divided into three main levels: User Level, Kernel Level, and Hardware Level. In the User Level, User Programs and Libraries interact with the System Call Interface. The Kernel Level contains the File Subsystem, Process Control Subsystem, Scheduler, and Memory management. It also includes a Buffer Cache and Device Drivers (character and block). The Hardware Level consists of Hardware Control and Hardware Level.

77

Μοντέρνος πυρήνας UNIX

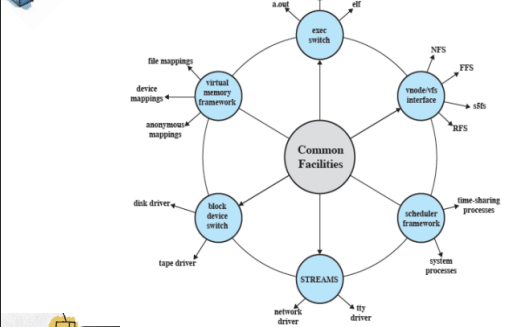
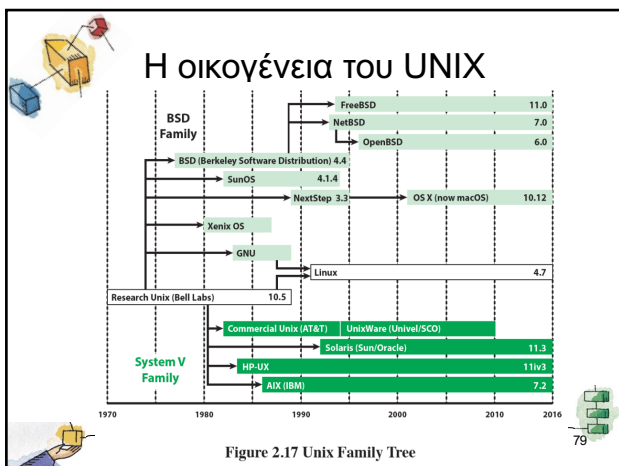


Figure 2.16 Modern UNIX Kernel

The diagram shows the modern UNIX kernel architecture. It features a central 'Common Facilities' block. Surrounding it are various components: 'virtual memory framework', 'block device switch', 'STREAMS', 'scheduler framework', 'time-sharing processes', 'system processes', 'network driver', 'tty driver', 'disk driver', 'tape driver', 'file mappings', 'device mappings', 'anonymous mappings', 'cdev switch', 'elf', 'a.out', 'NFS', 'FFS', 'ufs', and 'RFS'.

78



79

Περιεχόμενα

- Στόχοι και λειτουργίες του Λ.Σ.
- Η εξέλιξη των Λ.Σ.
- Κύρια επιτεύγματα.
- Εξελίξεις που οδήγησαν στα μοντέρνα Λ.Σ.
- Microsoft Windows.
- Συστήματα UNIX.
- - Linux.
- Android.

80

Ιστορική εξέλιξη του Linux

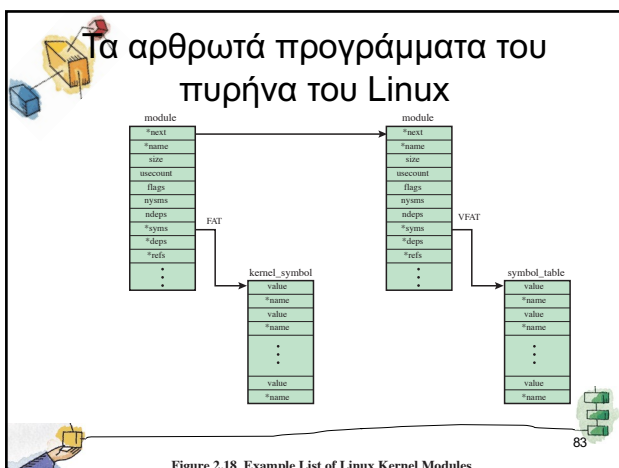
- Ξεκίνησε ως μία παραλλαγή του κλασικού Unix για προσωπικούς υπολογιστές (αρχικά το IBM PC).
- Ο πρώτη έκδοση γράφτηκε από τον [Linus Torvalds](#), ένα Φιλανδό φοιτητή.
- Εκδόθηκε στο διαδίκτυο για πρώτη φορά το 1991.
- Σήμερα είναι ένα πλήρως ανεπτυγμένο σύστημα Unix το οποίο υποστηρίζεται από διάφορες πλατφόρμες και του οποίου ο πηγαίος κώδικας είναι διαθέσιμος.
- Κλειδί της επιτυχίας του είναι η ύπαρξη και συνεχή ανάπτυξη ελεύθερων υπηρεσιών ως επεκτάσεις του.

81

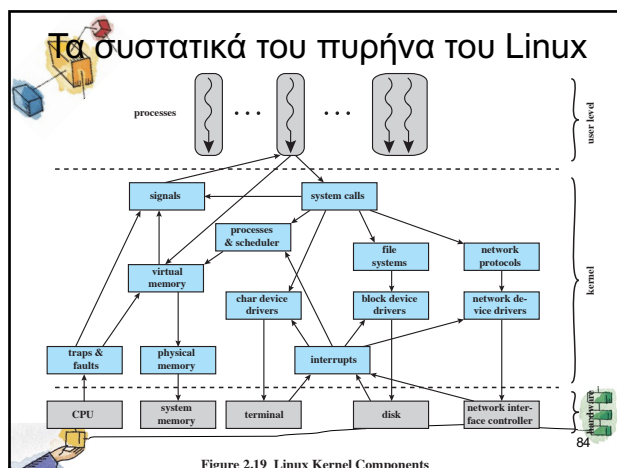
Αρθρωτός μονολιθικός πυρήνας

- Αν και μονολιθικός, ο πυρήνας είναι σχεδιασμένος ως μία ομάδα από αρθρωτά προγράμματα (modules).
- Στην ουσία, κάθε τέτοιο πρόγραμμα είναι ανεξάρτητο από τα άλλα και μπορεί να συνενωθεί με άλλα παρόμοια προγράμματα δυναμικά κατά την εκτέλεση των εφαρμογών.
- Επίσης, τα προγράμματα αυτά μπορούν να φορτωθούν στον πυρήνα δυναμικά ή να απομακρυνθούν από αυτόν.

82



83



84

Μερικά σήματα στο Linux

SIGHUP	Terminal hangup	SIGCONT	Continue
SIGQUIT	Keyboard quit	SIGTSTP	Keyboard stop
SIGTRAP	Trace trap	SIGTTOU	Terminal write
SIGBUS	Bus error	SIGXCPU	CPU limit exceeded
SIGKILL	Kill signal	SIGVTALRM	Virtual alarm clock
SIGSEGV	Segmentation violation	SIGWINCH	Window size unchanged
SIGPIPE	Broken pipe	SIGPWR	Power failure
SIGTERM	Termination	SIGRTMIN	First real-time signal
SIGCHLD	Child status unchanged	SIGRTMAX	Last real-time signal

85

Περιεχόμενα

- Στόχοι και λειτουργίες του Λ.Σ.
- Η εξέλιξη των Λ.Σ.
- Κύρια επιτεύγματα.
- Εξελίξεις που οδήγησαν στα μοντέρνα Λ.Σ.
- Microsoft Windows.
- Συστήματα UNIX.
- Linux.
- Android.

86

Εξέλιξη του Android

- Είναι βασισμένο στο Linux και αρχικά σχεδιάστηκε για κινητές συσκευές (π.χ. τηλέφωνα, ταμπλέτες) με οθόνες αφής.
- Έχει καταστεί το πιο δημοφιλές Λ.Σ. για κινητές συσκευές.
- Αναπτύχθηκε από την Android Inc. η οποία αγοράστηκε από τη Google το 2005.
- Η πρώτη εμπορική του έκδοση (Android 1) κυκλοφόρησε το 2008.
- Η πιο πρόσφατη έκδοση είναι το Android 11.
- Το 2007 δημιουργήθηκε ο οργανισμός Open Handset Alliance (OHA) από 84 εταιρείες με σκοπό την ανάπτυξη ανοικτών προτύπων για κινητές συσκευές.
- Το OHA είναι υπεύθυνο για την κυκλοφορία του Android ως ανοικτή πλατφόρμα, κάτι που αποτέλεσε κλειδί στην επιτυχία του συγκεκριμένου Λ.Σ.

87

Η αρχιτεκτονική λογισμικού του Android

The diagram illustrates the Android software architecture layers:

- Applications:** Home, Dialer, SMS/MMS, IM, Browser, Camera, Alarm, Calculator, Contacts, Voice Dial, Email, Calendar, Media Store, Albums, Clock, etc.
- Application Framework:** Activity Manager, Windows Manager, Content Providers, View System, Notification Manager, Package Manager, Telephony Manager, Resource Manager, Location Manager, SIMPP Service.
- System Libraries:** Surface Manager, Media Framework, SQLite, Core Libraries, OpenGL ES, FreeType, LibWebCore, Dalvik Virtual Machine, SSL, Libc.
- Linux Kernel:** Display Driver, Camera Driver, Bluetooth Driver, Flash Memory Driver, Binder (IPC) Driver, USB Driver, Keypad Driver, WiFi Driver, Audio Drivers, Power Management.

Implementation: Applications, Application Framework: Java; System Libraries, Android Runtime: C and C++; Linux Kernel: C.

88

Βασικά στοιχεία της αρχιτεκτονικής λογισμικού του Android

- Το πλαίσιο εφαρμογών παρέχει υψηλού επιπέδου συστατικά στοιχεία, τα οποία χρησιμοποιούνται μέσω API και βοηθούν στη δημιουργία νέων εφαρμογών.
- Η βιβλιοθήκη του συστήματος περιέχει μία σειρά από χρήσιμες ρουτίνες, υλοποιημένες σε C ή C++, οι οποίες χρησιμοποιούνται από διάφορα υποσυστήματα του Android και είναι προσβάσιμες μέσω του πλαισίου εφαρμογών.
- Το runtime μέρος του συστήματος διαχειρίζεται τις διεργασίες, η κάθε μία από τις οποίες εκτελείται στο δικό της χώρο (Dalvik ή ART).

89

Ο κύκλος ζωής ενός APK


The diagram shows the lifecycle of an APK:

- Source:** The starting point for development.
- Dex File:** Created from source code.
- APK:** The final package, created by zipping the Dex File and Resources & Native Code.
- Install:** The APK is installed on the device.
- Runtime:** The APK is processed by Dalvik (using dexopt to create ODEX File) or ART (using dex2oat to create ELF File).
- Execution:** The runtime environment (Dalvik or ART) executes the code, interacting with Native code and Libraries.

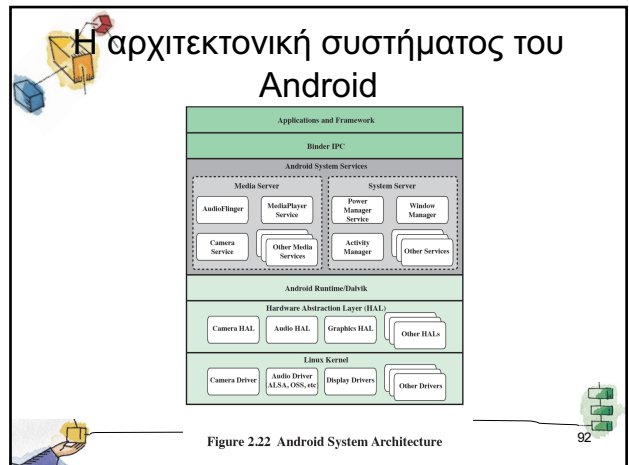
90

Το σύστημα Runtime του Android

- Συνήθως το λογισμικό για Android μεταφράζεται σε bytecode και μετά στον κώδικα μηχανής της κάθε συσκευής.
- Οι πρώτες εκδόσεις χρησιμοποιούσαν την τεχνική Dalvik, αλλά λόγω των περιορισμών σε χρήση μνήμης και εκμετάλλευση αρχιτεκτονικών multicore χρησιμοποιείται πλέον το Android Runtime (ART).
- Σε κάθε περίπτωση, μία εφαρμογή (APK) μεταφράζεται από πηγαίο κώδικα σε .dex και μετά εμπλουτίζεται με ότι επιπλέον κώδικα χρειάζεται για να δημιουργήσει ένα APK.




91



92

Βασικά χαρακτηριστικά της αρχιτεκτονικής συστήματος του Android

- Βασικά είναι μία απλοποιημένη μορφή της αρχιτεκτονικής λογισμικού, από το σημείο αναφοράς του δημιουργού μίας εφαρμογής.
- Με βάση αυτήν την αρχιτεκτονική, ο δημιουργός εφαρμογών επικεντρώνεται στη χρήση του στρώματος εφαρμογών και πλαισίου μέσω API.
- Το επίπεδο Binder IPC (Interprocess Communication) επιτρέπει την επικοινωνία μεταξύ του στρώματος εφαρμογών και του χαμηλότερου επιπέδου υπηρεσιών του πυρήνα του Λ.Σ.
- Αυτές οι υπηρεσίες επιτρέπουν την εκτέλεση πολυμέσων ή άλλων δραστηριοτήτων του συστήματος.



93



ΕΠΛ222: Λειτουργικά Συστήματα

(μετάφραση στα ελληνικά των διαφανειών του βιβλίου Operating Systems: Internals and Design Principles, 9/E, William Stallings)

Τέλος Ενότητας 1

Οι διαφάνειες αυτές έχουν συμπληρωματικό και επεξηγηματικό χαρακτήρα και σε καμία περίπτωση δεν υποκαθιστούν το βιβλίο

Γιώργος Α. Παπαδόπουλος
Τμήμα Πληροφορικής
Πανεπιστήμιο Κύπρου

94